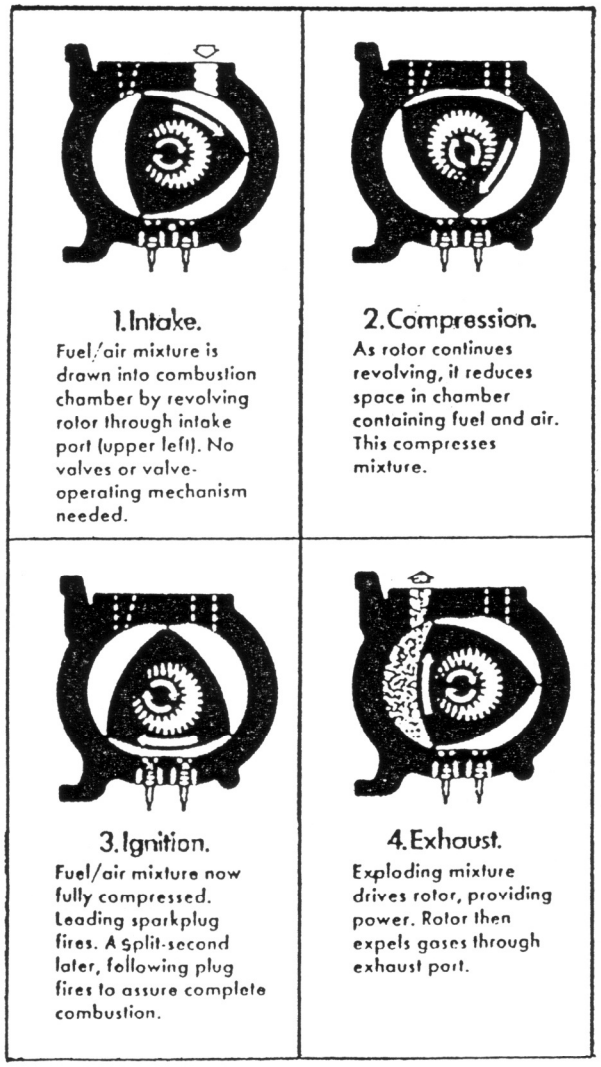
VZ-LINK MAGAZINE

NEW ZEALAND

VZ 300/200

COMPUTER USERS CLUB

COMPUTER MODELLING THE 'WANKEL' ENGINE



ISSUE 41

MARCH 90

Auto Engines of Tomorrow

FROM ME TO YOU

Welcome to another massive issue of VZ LINK.

In this issue we continue our Micro Magic series. I know we have covered extended basic several times in previous issues.
Micro Magic explains quite simply how to use these functions.
We also have a complete listing of Rattle Snake as promised.

We also have another item from Ken Clarke, this time we explore computer modelling using the Wankel rotary combustion engine as an example.

Last issue, we used a new printing company, I think the results where really good.

The questionnaire,s are very slow returning, PLEASE fill yours in and return as soon as possible. If for some reason, you didn't receive one or have lost it, contact me and I will send you another.

WARNING
Sometime ago Peter Patterson sent away an order for VZ Eproms to Gavin Williamson in Australia after seeing a advertisement in this newsletter. To date Peter has not heard from Gavin nor has repeated phone calls netted an answer. It would appear Peter could be out a couple of hundred dollars.
Please DO NOT order anything from Gavin until we find out what is going on.

Of interest is the appearance of several new VZ 300 in the Dick Smith stores. The origin of these VZ,s is unknown. D.S.E claim that they are new and were leftovers from the Aust. warehouse.
Price is $129.00 and there should be 2 in each store .

I am organising a Club meeting at the moment.
The date will be Saturday 7 April and a separate notice will be sent out within the next week, hopefully.
I am warning you now, so there can be no excuse for a poor attendance.

Until next issue, take care and keep on computing.

peter.

P.O.BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USER'S CLUB 1986

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT..
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT THEY ARE FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR THE BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME IF YOU HAVE PROBLEMS
PHONE 761-451

WANKEL Rotary Combustion Engine

- by Ken Clarke.

COMPUTER MODELLING

[An examination of the 'Wankel' Rotary Combustion Engine and a VZ-300 program to model its action.]

In theory a rotary engine is a much better proposition than a reciprocating engine since the purpose in both is to rotate a shaft. The German engineer, Dr. Felix Wankel developed the most successful rotary combustion engine. Wankel's rotary engine has only two principal moving parts, the Rotor and the shaft. As it rotates the rotor controls two ports, one for the intake of fresh fuel and the other for the removal of exhaust gases. The triangular shape of the rotor divides the (almost) figure of eight shape of the engine interior into three chambers. The turning rotor is continually increasing and decreasing the size of these three 'moving chambers', each one being an analogue of the cyclinder in a standard piston engine.

As the rotor rotates, the volume of one chamber becomes smaller in a compression phase. When the volume is a minimum the spark plug fires to initiate the combustion phase. The volume of that chamber then increases up to the point where the exhaust port is uncovered. Its volume then decreases to chase the spent fuel out of the chamber. As soon as the exhaust phase is completed and the exhaust gases have been expelled from the chamber then that chamber moves over the inlet port and its volume increases once more so that fresh fuel mixture can be taken in before the next compression phase begins.

The three chambers are always at different stages of the four stroke cycle and each of the three chambers completes a full cycle of changes while the rotor turns through one 360 degree revolution. While one chamber may be on the intake phase, the compression and the ignition phases take place in another chamber and the third chamber will be on the exhaust phase.

The rotor moves in a circular orbit around the output shaft and as a consequence each complete revolution of the rotor results in three revolutions of the output shaft.

MINOR PROBLEMS

Rotary engines suffer from a tendency to leave some of the fuel mixture unburned. This reduces the engine efficiency and also produces an undesirable source of atmospheric pollution. Another problem is concerned with the need to ensure that the three combustion chambers remain gas tight. Seals at the rotor tips and faces can largely overcome these problems.

The Wankel Rotary engine has several advantages. It has fewer moving parts and is lighter and more compact than comparable piston engines and has a superior power to weight ratio. The rotor and the output shaft both rotate in the same direction and at constant speeds. This has the effect of reducing engine vibrations and also makes the Wankel engine markedly quieter than a reciprocating engine. The Japanese automobile manufacturer, Mazda, has used Wankel engines in some of its vehicles over the past few years.

PROGRAM DESCRIPTION

The 'active' part of the VZ-300 computer program, to model the Wankel rotary engine, is really quite short. The animation to show the engine operating, as described above, is created by using four separate rotor and chamber pictures. These are made to follow each other in rapid succession to produce the effect of continuous rotor movement. Subroutine 1000 is used to build up these pictures which are held in the string array P$. Individual lines of each picture are first held in another string array, L$, and each picture is built up from the individual lines (1120/1130, 1220/1230, 1320/1330, 1420/1430). This method of building up the large P$ strings from the smaller L$ strings is used to make the entry and editing of the graphics easier to handle.

The static part of the display is printed out at the end of the subroutine starting at line 1000. With the picture array established, the loop F (100-190) is used to control the changing display. Within the F loop there is a smaller loop which is used to show the rotation of the shaft using the four graphic characters held in array A$. The program waits at line 110 whenever INKEY$ detects that a key is being depressed to give the user control over the action. The program continues as soon as the key is released.

```
1 CLS:COLOR3,1:'"▌▌▌▌BY▌KEN▌CLARKE▌▌▌AUG▌1989▌▌▌▌"
10 FORL=1TO64:PRINT"WANKEL ";:NEXTL
20 PRINTTAB(3);"ROTARY▌COMBUSTION▌ENGINE"
30 FORL=1TO90:PRINT"      ";:NEXTL
50 CLEAR1350:GOSUB1000
60 X=1
100 FORF=1TO7STEP2
110 IFINKEY$<>"",110
120 PRINT@102,P$(F)P$(F+1)
122 IFF=1,PRINT@184,"▌▌TAKE"
124 IFF=3,PRINT@373,"COMPRESSION"
125 IFF=5,PRINT@354,"IGNITION"
126 IFF=5,PRINT@400,"▌":PRINT@459,"SPARK▌PLUG"
128 IFF=7,PRINT@400,"=":PRINT@459,"SPARK-PLUG"
130 IFF=7,PRINT@131,"EXHAUST"
140 FORG=1TO3
145 PRINT@240,A$(X)
150 X=X+1
155 IFX=5,X=1
160 NEXTG
190 NEXTF
200 GOTO100
1000 DIMP$(8)
1010 DIML$(11)
1020 DIMS$(14)
1030 DIMA$(4)
1032 A$(1)="▟"                                          :R
1034 A$(2)="▙"                                          :E
1036 A$(3)="▛"                                          :Q
1038 A$(4)="▜"                                          :W
1102 L$(2)="  ▐▌▗          ▗▟▌                        " :2s,U,▌,A,10s,E,2▌,14s
1103 L$(3)="  ▌▗▛▀▀ ▌▌▌▌▌▌▌▌ ▀▀▙▌▌▌                  " :2s,▌,A,Q,T,F,6▌,D,T,W,E,2▌
1104 L$(4)=" ▀▗▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▗▌▌▌▗              "
1105 L$(5)=" ▌▌▗ ▌▌▌▌▌▌▗ ▗ ▌▌▌▌▌▗ ▌▌▌                "
1106 L$(6)="▗▌▌  ▗▌▌▌▌▗   ▗▌▌▌▌▗  ▗▌▌▗               "
1107 L$(7)=" ▌▌▗ ▙▌▌▌▌▌▀ ▀▌▌▌▗   ▌▌▌                "
1108 L$(8)=" ▗ ▀▗ ▗▌▌▌▌▌▌▌▌▗   ▀▌▗                  "
1109 L$(9)=" ▗▌▀▗ ▗▟▌▌▌▌▌▗   ▀▀▌▌                   "
1110 L$(10)="  ▗▌▀▗ ▙▗▌▗  ▀▀▌▗                        "
1120 P$(1)=L$(2)+L$(3)+L$(4)+L$(5)+L$(6)+L$(7)
1130 P$(2)=L$(8)+L$(9)+L$(10)
```

NOTES on GRAPHICS

The codes after the graphics give the number of spaces or graphics and the keys, where to find the graphics.

Only a short sample has been given to get you started.

(: Line 1540 is - E,1,R)

```
1202 L$(2)="
1203 L$(3)="
1204 L$(4)="
1205 L$(5)="
1206 L$(6)="
1207 L$(7)="
1208 L$(8)="
1209 L$(9)="
1210 L$(10)="
1220 P$(3)=L$(2)+L$(3)+L$(4)+L$(5)+L$(6)+L$(7)
1230 P$(4)=L$(8)+L$(9)+L$(10)
1302 L$(2)="
1303 L$(3)="
1304 L$(4)="
1305 L$(5)="
1306 L$(6)="
1307 L$(7)="
1308 L$(8)="
1309 L$(9)="
1310 L$(10)="
1320 P$(5)=L$(2)+L$(3)+L$(4)+L$(5)+L$(6)+L$(7)
1330 P$(6)=L$(8)+L$(9)+L$(10)
1402 L$(2)="
1403 L$(3)="
1404 L$(4)="
1405 L$(5)="
1406 L$(6)="
1407 L$(7)="
1408 L$(8)="
1409 L$(9)="
1410 L$(10)="
1420 P$(7)=L$(2)+L$(3)+L$(4)+L$(5)+L$(6)+L$(7)
1430 P$(8)=L$(8)+L$(9)+L$(10)
1450 COLOR1,0
1500 PRINT@72,"
1530 PRINT@393,"
1540 PRINTTAB(15);"
1550 PRINT@480,"PRESS ANY KEY TO STOP THE ROTOR";
1600 RETURN
```


SURE SERVICE SPECIAL SERVICE

QUALITY QUICK PRINTING

PRINTING FOR
- YOU
- YOUR COMPANY
- YOUR TEAM

OFFER YOU
- PERSONAL & PROFICIENT SERVICE
- EXCELLENCE IN THE '90's
- OUTSTANDINGLY COMPETITIVE RATES

SUPER PRINT

SERVICE • QUALITY • PRICE

523 PARNELL RD. PARNELL
P.O. BOX 99-085 NEWMARKET
☎ (09) 370-900 • FAX (09) 372-216

370-900 • 370-900 • 370-900

rattle snake

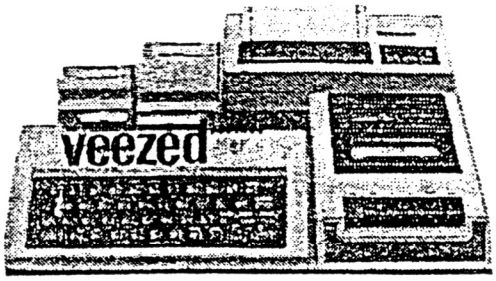
a full program listing

```
50 '    *****************
60 '    * MICRO MAGIC 3 *
70 '*********************
80 '* RATTLESNAKE *
90 '***************
100 CLS:CLEAR:GOSUB10000:GOSUB3000:GOTO2000
1000 IFN=1ANDA=194THENL=L+10000
1001 IFN=1ANDA=200THENL=L+1000
1002 IFN=1ANDA=206THENL=L+100
1003 IFN=1ANDA=212THENL=L+10
1004 IFN=1ANDA=218THENL=L+1
1010 FORS=0TO4:COLOR4:PRINT@A,"▀▀▀▀▀▀▀▀":A=A+32:NEXT:A=A-160:COLOR5
1020 PRINT@A,      "      "'ZZZZZ
1021 PRINT@A+32,   " ▄▄▄▄ "'ZJJJZ
1022 PRINT@A+64,   " ▄▄▄▄ "'ZJJJZ
1023 PRINT@A+96,   "      "'ZZZZZ
1024 PRINT@A+128,  " ▄▄▄▄ "'ZJJJZ
1030 RETURN
1040 IFN=1THEN:GOSUB2000'  ACE
1050 IFN=2THEN:GOSUB2100'  TEN
1060 IFN=3THEN:GOSUB2200' JACK
1070 IFN=4THEN:GOSUB2300'QUEEN
1080 IFN=5THEN:GOSUB2400' KING
1090 IFN=6THEN:GOSUB2500'SNAKE
1100 IFN=2ANDA=194THENL=L+20000
1101 IFN=2ANDA=200THENL=L+2000
1102 IFN=2ANDA=206THENL=L+200
1103 IFN=2ANDA=212THENL=L+20
1104 IFN=2ANDA=218THENL=L+2
1110 FORS=0TO4:COLOR4:PRINT@A,"▀▀▀▀▀▀▀▀":A=A+32:NEXT:A=A-160:COLOR5
1120 PRINT@A,     " ▄▄ ▄▄▄ "'ASYYS
1121 PRINT@A+32,  " ▄▌ ▐ "'IUUZU
1122 PRINT@A+64,  " ▄▌ ▐ "'IUUZU
1123 PRINT@A+96,  " ▄▌ ▐ "'IUUZU
1124 PRINT@A+128," ▄▄ ▄▄▄ "'DFTTF
1125 RETURN
1130 RETURN
1140 PRINT@71,W:PRINT@88,C
1150 W=0:GOTO3050
1200 IFN=3ANDA=194THENL=L+30000
1201 IFN=3ANDA=200THENL=L+3000
1202 IFN=3ANDA=206THENL=L+300
1203 IFN=3ANDA=212THENL=L+30
1204 IFN=3ANDA=218THENL=L+3
1210 FORS=0TO4:COLOR4:PRINT@A,"▀▀▀▀▀▀▀▀":A=A+32:NEXT:A=A-160:COLOR6
1220 PRINT@A,     "▄▄▄▄▄▄"'JJJTW
1221 PRINT@A+32,  "▄▄▄▄  ▌"'JJJZI
1222 PRINT@A+64,  "▄▄▄▄  ▌"'JJJZI
1223 PRINT@A+96,  "▌ ▀▀  ▌"'UDTZI
1224 PRINT@A+128,"▌▄▄▄▄▄▌"'EYYYR
1230 RETURN
1300 IFN=4ANDA=194THENL=L+40000
1301 IFN=4ANDA=200THENL=L+4000
1302 IFN=4ANDA=206THENL=L+400
1303 IFN=4ANDA=212THENL=L+40
1304 IFN=4ANDA=218THENL=L+4
```

```
1310 FORS=0TO4:COLOR4:PRINT@A,"▬▬▬▬":A=A+32:NEXT:A=A-160:COLOR2
1320 PRINT@A,     "▛▜▜▜▜▜▜▜"'QTTTW
1321 PRINT@A+32, "▌ ▐▐▐▐ ▐"'UIJUI
1322 PRINT@A+64, "▌ ▐▐▐▐ ▐"'UIJUI
1323 PRINT@A+96, "▌ ▝▝ ▐"'UDFFI
1324 PRINT@A+128,"▐▐▐▐ ▜"'JJUTW
1330 RETURN
1400 IFN=5ANDA=194THENL=L+50000
1401 IFN=5ANDA=200THENL=L+5000
1402 IFN=5ANDA=206THENL=L+500
1403 IFN=5ANDA=212THENL=L+50
1404 IFN=5ANDA=218THENL=L+5
1410 FORS=0TO4:COLOR4:PRINT@A,"▬▬▬▬":A=A+32:NEXT:A=A-160:COLOR7
1420 PRINT@A,     "▛▜▛▜▜▜▜"'QWJTW
1421 PRINT@A+32, "▌▛▗▟▟▟▟"'JIFRJ
1422 PRINT@A+64, "▌▗▜▜▜▜"'JADJJ
1423 PRINT@A+96, "▌▌▙▜▜▜"'JIEDJ
1424 PRINT@A+128,"▙▟▟▟▟▟"'EYJER
1430 RETURN
1500 IFN=6ANDA=194THENL=L+60000
1501 IFN=6ANDA=200THENL=L+6000
1502 IFN=6ANDA=206THENL=L+600
1503 IFN=6ANDA=212THENL=L+60
1504 IFN=6ANDA=218THENL=L+6
1510 FORS=0TO4:COLOR4:PRINT@A,"▬▬▬▬":A=A+32:NEXT:A=A-160:COLOR2
1520 PRINT@A,     "▛▛▟▟▟▟"'QQEYY
1521 PRINT@A+32, "▝▝ ▟▟"'DTZYR
1522 PRINT@A+64, "▝▗▟▙▙"'FYSEY
1523 PRINT@A+96, "▝▌▙▟▟"'FUEYI
1524 PRINT@A+128,"▙▟  ▟▟"'EUZIR
1530 RETURN
1600 IFN=7ANDA=194THENL=L+70000
1601 IFN=7ANDA=200THENL=L+7000
1602 IFN=7ANDA=206THENL=L+700
1603 IFN=7ANDA=212THENL=L+70
1604 IFN=7ANDA=218THENL=L+7
1610 FORS=0TO4:COLOR4:PRINT@A,"▬▬▬▬":A=A+32:NEXT:A=A-160:COLOR8
1620 PRINT@A,     "▟▟▟▟▟▟"'AYYYS
1621 PRINT@A+32, "▌▟▟▟▟▟"'IAYSU
1622 PRINT@A+64, "▟▟▟▟▟▟"'JOKER
1623 PRINT@A+96, "▌▝▝▝▝▝"'IDTFU
1624 PRINT@A+128,"▟▟▟▟▟▟"'DTTTF
1630 RETURN
2000 IFC=0THEN3000
2010 PRINT@71,W:PRINT@88,C
2020 COLOR1:C=C-1:PRINT@71,"▟▟▟▟▟▟▟▟":PRINT@88,"▟▟▟▟▟":PRINT@88,C
2030 FORI=1TORND(5):L=0:A=194:FORJ=1TO5:N=RND(7)
2040 IFN=1THENGOSUB1000'   ACE
2042 IFN=2THENGOSUB1100'   TEN
2044 IFN=3THENGOSUB1200'  JACK
2046 IFN=4THENGOSUB1300' QUEEN
2048 IFN=5THENGOSUB1400'  KING
2050 IFN=6THENGOSUB1500' SNAKE
2052 IFN=7THENGOSUB1600' JOKER
2060 A=A+6:NEXT
2070 A=A+32:NEXT
2080 SOUND31,1:T=0:GOSUB4000
2090 PRINT@71,W:PRINT@88,C:W=0:GOTO3050
2999 'INPUT COINS
3000 FORI=384TO415:PRINT@I," ":NEXT
3010 PRINT@392,"▟▟▟▟▟▟▟▟
```

```
3049 ' PULL HANDLE
3050 IFC=0THEN3000
3060 Z$=INKEY$:Z$=""
3070 Z$=INKEY$:IFZ$=""THEN3060
3080 IFZ$="P"THEN2000ELSE3060
3999 ' PAYS FOR WINS
4000 IFL=11111THENW=2000:C=C+W:GOSUB8000:RETURN
4002 IFL>11110ANDL<11118THENW=1000:GOSUB8000:RETURN
4004 IFL>11100ANDL<11188THENW=500:GOSUB8020:RETURN
4006 IFL=22222THENW=50:C=C+W:SOUND30,6;18,4;30,4;18,8:RETURN
4008 IFL>22220ANDL<22228THENW=20:C=C+W:SOUND31,6;19,8:RETURN
4010 IFL>22200ANDL<22288THENW=10:C=C+W:SOUND31,8:RETURN
4012 IFL>22000ANDL<22888THENW=5:C=C+W:SOUND18,4:RETURN
4014 IFL>20000ANDL<28888THENW=2:C=C+W:SOUND16,2:RETURN
4016 IFL=33333THENW=400:C=C+W:GOSUB8000:RETURN
4018 IFL>33330ANDL<33338THENW=200:C=C+W:GOSUB8000:RETURN
4020 IFL>33300ANDL<33388THENW=100:C=C+W:SOUND31,6;19,8:RETURN
4022 IFL=44444THENW=400:C=C+W:GOSUB8000:RETURN
4024 IFL>44440ANDL<44448THENW=200:C=C+W:GOSUB8000:RETURN
4026 IFL>44400ANDL<44488THENW=100:C=C+W:SOUND31,6;19,8:RETURN
4028 IFL=55555THENW=500:C=C+W:GOSUB8020:RETURN
4030 IFL>55550ANDL<55558THENW=300:C=C+W:GOSUB8000:RETURN
4032 IFL>55500ANDL<55588THENW=100:C=C+W:SOUND31,6;19,8:RETURN
4034 IFL=77777THENW=1000:C=C+W:GOSUB8000:RETURN
4036 IFL>77770ANDL<77778THENW=5000:C=C+W:GOSUB8020:RETURN
4038 IFL>77700ANDL<77788THENW=200:C=C+W:GOSUB8010:RETURN
4040 IFL=66666THENW=10000:C=C+W:GOSUB8030:RETURN
4042 IFL>66660ANDL<66668THENW=5000:C=C+W:GOSUB8000:RETURN
4044 IFL>66600ANDL<66688THENW=200:C=C+W:GOSUB8020:RETURN
4046 IFL>66000ANDL<66888THENW=50:C=C+W:GOSUB8020:RETURN
4048 IFL>60000ANDL<68888THENW=20:C=C+W:SOUND30,4;18,8:RETURN
4050 RETURN
7999 'JACKPOTS
8000 PRINT@427,"[illegible]":PRINT@71,W:PRINT@88,C:GOSUB9000
8002 FORI=29099TO29106:POKEI,143:NEXT:RETURN
8010 PRINT@427,"[illegible]":PRINT@71,W:PRINT@88,C:GOSUB9020
8012 FORI=29099TO29106:POKEI,143:NEXT:RETURN
8020 PRINT@427,"[illegible]WOW![illegible]":PRINT@71,W:PRINT@88,C:GOSUB9020
8022 FORI=29099TO29106:POKEI,143:NEXT:RETURN
8030 PRINT@384,"[illegible]"
8040 PRINT@71,W:PRINT@88,C
8050 PRINT@390,"[illegible]":GOSUB9000
8052 GOSUB10000:RETURN
8999 'MUSICAL FANFARES
9000 SOUND16,2;18,2;20,4;20,2;20,2;18,2;0,2;16,2;18,2;20,2;16,2
9002 SOUND13,2;15,2;16,2;0,2;11,2;16,2;20,2;23,4;21,2;20,2;18,4
9004 SOUND18,2;18,2;16,6:RETURN
9020 SOUND13,1;14,1;15,1;16,1;17,1;18,1;19,1;20,1;21,1;22,1;23,1
9030 SOUND28,1;27,1;25,1;23,1;21,1;20,1;18,1;16,1;28,2;16,4
9050 :RETURN
9999 '"[illegible]
10000 COLOR3,1
10010 PRINT@0,"[illegible]R[illegible]A[illegible]T[illegible]T[illegible]L[illegible]E[illegible]S[illegible]N[illegible]A[illegible]K[illegible]E[illegible]":COLOR8
10020 FORI=28704TO29183:POKEI,143:NEXT
10030 COLOR8:FORI=160TO191:PRINT@I,"[illegible]":NEXT
10040 B=192:FORI=1TO5:PRINT@B,"[illegible]  [illegible]  [illegible]  [illegible]  [illegible]  [illegible]"
10050 B=B+32:NEXT
10060 FORI=352TO383:PRINT@I,"[illegible]":NEXT
10070 PRINT@67,"[illegible]":PRINT@81,"[illegible]"
10080 RETURN
```

MICRO MAGIC WORKSHOP



COMPATIBILITY

At first, asterisks in DSE advertisements suggested that the VZ-200 16K Expansion Module was compatible with the VZ-300. However, such was not the case. Whereas the VZ-200 had only 6K of in-built memory, the VZ-300 has 16K. This means that the first memory address of the VZ-200 module does not follow on from the last memory address of the VZ-300.

The VZ-300 goes to 63487 and the VZ-200 module begins at 55248, which means that the only part of the module usable is from 63488 to the top of the Module's memory, about a mere 1K.

Now, if you plug a VZ-300 Expansion Module into a VZ-200, and we cannot imagine why you should, there would be a gap between the highest address of the computer and the lowest address of the module. The computer would ignore the module altogether.

In the case of the VZ-300's 64K Module (if you were ever lucky enough to obtain one), this too presents a problem. The VZ-300 can only accommodate a total of 34K. A system known as BANK SWITCHING puts three banks of 16K in parallel with a fourth bank of 16K available for instant use. The three other banks all have the same memory addresses and can only be selected using Machine Code,not BASIC.

(See previous issues of VZ-LINK for how to access 64K Modules, peter)

Of interest is the plug-in word-processor for the VZ-300. It allows the Expansion Module to be piggy-backed and thus provide a very useful package.

Then, of course, by adding a Disk Drive you obtained the advantages of fast loading and storage (about 14 times faster than cassette). The Disk has storage capacity of about 78K and the Disk commands are built-in with the help of a Disk Controller module.

What a pity the designers blocked off so many useful commands such as: ON...GOTO/GOSUB, DEFINT, VARPTR, FRE etc. with all those extra commands the VZ with Disk Drive would be superior to computers costing many times more. In Studio Ad Lib we show you how to access more than twenty of the INVISIBLE commands.

TRON = TRACE ON TROFF = TRACE OFF

The best way of describing what TRON and TROFF do is to demonstrate.....

Enter and Run:

```
10 CLS:PRINT@320,"MICRO MAGIC"
20 FOR I=1 TO 10:SOUND 16,3;18,6:NEXT
30 COTO 30
```

Now add Line 5 POKE 31003,175 and run again....

What you see is your line numbers being printed as they are being RUN. It is up to you how to use TRON when debugging a program. Obviously, the closer to the bug the better.
To turn TRON off (TROFF) enter POKE 31003,0

TOKENS

TOKEN	COMMAND	TOKEN	COMMAND	TOKEN	COMMAND	TOKEN	COMMAND
128	END	160	OUT	192	VARPTR	224	EXP
129	FOR	161	ON	193	USR	225	COS
130	RESET	162		194	ERL	226	SIN
131	SET	163		195	ERR	227	TAN
132	CLS	164		196	STRING$	228	ATN
133		165		197		229	PEEK
134	RANDOM	166		198	POINT	230	
135	NEXT	167		199		231	
136	DATA	168		200	MEM	232	
137	INPUT	169		201	INKEY$	233	
138	DIM	170		202	THEN	234	
139	READ	171		203	NOT	235	
140	LET	172		204	STEP	236	
141	GOTO	173		205	+	237	
142	RUN	174	(RESET)	206	-	238	
143	IF	175	LPRINT	207	*	239	CINT
144	RESTORE	176		208	/	240	CSNG
145	GOSUB	177	POKE	209	^	241	CDBL
146	RETURN	178	PRINT	210	AND	242	FIX
147	REM	179	CONT	211	OR	243	LEN
148	STOP	180	LIST	212	>	244	STR$
149	ELSE	181	LLIST	213	=	245	VAL
150	COPY	182	DELETE	214	<	246	ASC
151	COLOR	183	AUTO	215	SGN	247	CHR$
152	VERIFY	184	CLEAR	216	INT	248	LEFT$
153	DEFINT	185	CLOAD	217	ABS	249	RIGHT$
154	DEFSNG	186	CSAVE	218	FRE	250	MID$
155	DEFDBL	187	NEW	219	INP	251	
156	CRUN	188	TAB(	220	POS	252	
157	MODE	189	TO	221	SQR	253	
158	ERROR	190		222	AND	254	
159	RESUME	191	USING	223	LOG	255	

In any BASIC program, the first memory location after the line number is 31469

This is regardless of what the line number is. All line numbers take up the same amount of memory. If the first command is REM and we POKE 31469,132 it will change REM into CLS.

If a COMMAND not in the vocabulary of the Vee Zed is POKEd into 31469, the line will appear empty. The TOKEN is, however, firmly in place and the command will be executed.

These are the commands we call INVISIBLE COMMANDS

RANDOM 10# POKE 31469,134
Reseeds RND(n) to give different series of RANDOM numbers.

DEFINT 10#A,B POKE 31469,153
Define all variables beginning with A and B as INTEGERS.

DEFSNG 10#C,D POKE 31469,154
Defines all variables beginning with C and D as SINGLE PRECISION (6/7 floating point)

DEFDL 10#E,F POKE 31469,155
Defines all variables beginning with E and F as DOUBLE PRECISION (16/17 floating point)

ON A GOTO 100,200,300 POKE 31469,161
10 #A GOTO 100,200,300

ON A GOSUB 100,200,300 POKE 31469,161
10 #A GOSUB 100,200,300

ON ERROR GOTO 1000 ON=POKE 31469,161
10## GOTO 1000 ERROR=POKE 31470,158

ON ERROR GOSUB 1000 ON=POKE 31469,161
10## GOSUB 1000 ERROR=POKE 31470,158

RESUME 10# POKE 31469,159
After ERROR return to error point.
10# 100 POKE 31469,159
After ERROR GOTO 100
10# 100 NEXT POKE 31469,159
After ERROR return to line AFTER error point.

AUTO 10# POKE 31469,183
Automatically prints line numbers beginning at 10 incrementing by 10s.
10# 100,10 POKE 31469,183
Automatically prints line numbers beginning at 100 incrementing by 10s.

VARPTR 10# (X) POKE 31469,192
Used to locate memory address of a Variable.

STRING$
10 PRINT#(12,45) POKE 31470,196
Will PRINT 12 Asterisks (ASCII 45)
Maximum length of string is 256 char.

MEM 10 PRINT# POKE 31470,200
Returns the amount of unused memory.

FRE 10 PRINT#(A) POKE 31470,218
Returns the number of unused BYTES.
10 PRINT#(A$) POKE 31470,218
Returns the number of unused BYTES in the reserved String spaces.

CINT 10 #Z POKE 31469,239
Removes all digits after the decimal point.

CSNG 10 #Z POKE 31469,240
Converts numeric variable from DOUBLE to SINGLE PRECISION.

CDBL 10 #Z POKE 31469,241
Converts numeric variable from SINGLE to DOUBLE PRECISION.

FIX 10 A=#(n) POKE 31472,242
Removes all digits to the right of the decimal point.Does not round-off negative numbers.

ERL 10 PRINT# POKE 31470,194
Returns the line number from which a program branched to ERROR routine.

ERR 10 PRINT# POKE 31470,195
Returns value related to type of ERROR

POS 10 A=#(X) POKE 31471,220
Returns current CURSOR position.
(X) is dummy argument.

NOTE If AUTO is halted by BREAK, POKE 30945,175 to restart AUTO from where it was halted.

Suppose you want to use MEM. This is the command for returning the amount of unused memory. The following are three methods of putting the TOKEN for MEM into a program. As MEM must be preceded by the command PRINT the address for PRINT must be 31469, therefore MEM must be POKEd into 31470. To do this we reserve that address by typing in a DUMMY CHARACTER before making the POKE.The DUMMY CHARACTER used here is the HASH sign, # .

FIRST LINE METHOD

```
10 PRINT#       (return)
POKE 31470,200 (return)
```

(CONTINUED ON NEXT PAGE)

When LISTed line 10 appears as...10 PRINT The command is invisible but the token for MEM is in place and when RUN will give the unused memory space.

GOSUB METHOD

MEM in the first line may be inconvenient.
This method lets you use MEM anywhere in your program. The line to receive the token must be entered first, as....

```
20 PRINT#        (return)
POKE 31470,200   (return)
```

now enter.

```
10 GOTO 100      (return)
30 RETURN        (return)
```

Line 100 now becomes the true beginning of the program and GOSUB20 can be used anywhere, Whenever MEM is required.

PEEK ADDRESS METHOD

To put MEM anywhere in a program requires the determining of the EXACT address. While a program is being entered, at the spot where MEM is required, enter the dummy character #.

The program should be correct up to this point.
Let's assume the last line you typed into the incomplete program is....

```
1130 CLS:PRINT"MEMORY";:#
```

Now enter directly (without line number) the following...

```
PRINT PEEK(30969)+256*PEEK(30970)-40
```

The number returned is the address of #.

POKE this number with 200 (for MEM). If the number is greater than 32767 subtract 65536.
You can now continue entering your program from the NEXT line number.
CARE must be taken not to overwrite the invisible MEM.

"The 11th Commandment"

FINDING YOURSELF IN MEMORY

Enter the following....

```
10 PRINT"ALPHABET"
20 FOR S=31410 TO 31510
30 C$=CHR$(PEEK (S))
40 IF C$<> "A" THEN 80
50 D$=CHR$(PEEK(S+1))
60 IF D$<> "L" THEN 80
70 PRINT S
80 NEXT
100 INPUT"STARTING LOCATION";X
110 FOR Z=X TO X+7
120 PRINT Z;CHR$(PEEK(Z))
130 NEXT
```

Now type RUN (return)

Line 10 is DUMMY TEXT.
Line 20 contains Memory Locations
Lines 30 and 40 allow the LOOP to run until the letter A is found.
Lines 50 and 60 checks the next locations for the letter L.
If both an A and a L are found then the memory location of A is printed. It may not be the word ALPHABET in which case the search will have to be restarted using other memory locations in the loop. In this example the locations are correct and the following is what you should have....

```
ALPHABET
 31471
STARTING LOCATION?  you enter 31471
 31471 A
 31472 L
 31473 P
 31474 H
 31475 A
 31476 B
 31477 E
 31478 T
```

Instructions, such as PRINT, are coded. The code is called a TOKEN. The TOKEN for PRINT in the above experiment must be just before 31471. Agree? So, RUN again, but this time enter 31467 and you will see that THAT location contains ASCII 178 (SHIFT Z). This is the TOKEN for PRINT.

Now that we know WHERE the first line of the program is stored, and we know WHAT the TOKEN for PRINT is, making use of this knowledge could be your first step into Machine Code programming.

FOR, TO, NEXT, RETURN, THEN, Etc., Etc., are COMMANDS (sometimes called FUNCTIONS). They tell the computer to DO something. Every command has a TOKEN, which is similar to ASCII numbers, except that the computer reads them as commands.
FINDING YOURSELF IN MEMORY, I thought, would help you to understand what TOKENS were all about. In the list of Tokens included with this issue, those that are not accessable from the keyboard are underlined. They are easily used. We simply have to POKE them into our programs. These commands are built-in to the computer just as PRINT and LIST and all those others on the keyboard.
The difference is that their names, such as AUTO and DELETE are not in the computers vocabulary.

Let MICRO MAGIC show you just how EASY it is to use them.

CONSIDER THIS...

```
10 PRINT"ELEPHANT"
20 FOR A=31469 TO 31479
30 READ B:POKEA,B
40 NEXT
50 DATA 178,34,67,79,77,80,85,84,69,82,34
```

Now type RUN (return)

```
ELEPHANT
```

Now type RUN (return) again

```
COMPUTER
```

Now LIST Line 10

```
10 PRINT"COMPUTER"
```

Whatever happened to the ELEPHANT ???